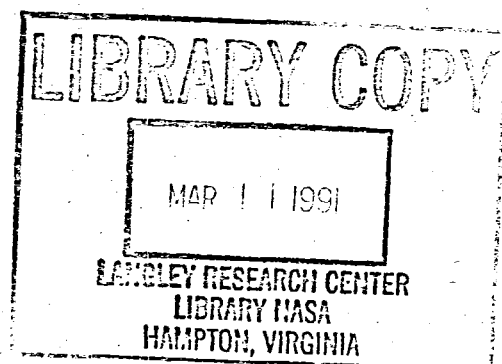


**NASA
Technical
Paper
3069**

March 1991

Structural Factoring Approach for Analyzing Stochastic Networks

Kelly J. Hayhurst
and Douglas R. Shier



NASA

1991

Structural Factoring Approach for Analyzing Stochastic Networks

Kelly J. Hayhurst
Langley Research Center
Hampton, Virginia

Douglas R. Shier
The College of William and Mary
Williamsburg, Virginia



National Aeronautics and
Space Administration
Office of Management
Scientific and Technical
Information Division

Contents

Symbols	iv
Abstract	1
Introduction	1
Notation	2
Structural Reduction Techniques	2
Fundamental Reductions	3
Conditioned Reductions	5
Conditioned Reductions and Shortest Path Calculations	10
Total Factoring	13
Determining the Shortest Path Length Distribution	13
Applications of the Structural Factoring Approach	15
Critical Path Analysis	15
Two-Terminal Reliability	16
Comparison Between Structural Factoring and Complete Enumeration	18
Conclusions	18
References	19

Symbols

A	set of arcs in a directed graph
$a_{(i,j)}$	n -tuple of lengths for arc (i, j)
c_j	j th length in an n -tuple
$f(i)$	factoring arc of node i
$f_i(j)$	i th factoring arc of node j
G	directed graph
(i, j)	arc with origin node i and destination node j
$\text{indegree}(i)$	number of arcs entering node i
$k(i, j)$	constant length for arc (i, j)
$L(i, j)$	particular length for arc (i, j)
m	number of arcs in the network
N	set of nodes in a directed graph
n	number of nodes in the network
$\text{outdegree}(i)$	number of arcs leaving node i
$P()$	probability function
$p_{(i,j)}$	probability distribution for the lengths of arc (i, j)
S_j	subnetwork j
s	source node
t	terminal node
γ	number of factoring arcs
$\omega(i)$	number of subnetworks generated by factoring on node i
$ (i, j) $	number of lengths for arc (i, j)

Abstract

In this report, the problem of finding the distribution of the shortest path length through a stochastic network is investigated. A general algorithm is developed for determining the exact distribution of the shortest path length. The algorithm is based on the concept of conditional factoring, in which a directed, stochastic network is decomposed into an equivalent set of smaller, generally less complex sub-networks. Several network constructs are identified and exploited so that the computational effort required to solve a network problem is significantly less than that required by complete enumeration. This algorithm can be applied to two important classes of stochastic path problems: determining the critical path distribution for acyclic networks and the exact two-terminal reliability for probabilistic networks. Computational experience with the algorithm has been encouraging and has allowed the exact solution of networks previously analyzed only by approximation techniques.

Introduction

In general, network analysis involves the study of systems that can be modeled in terms of nodes and arcs connecting certain pairs of nodes. Quantitative information, such as length, reliability, or time to completion, may also be associated with the arcs. For example, figure 1 shows a simple six-node network representing possible shipping routes from a factory (node A) to a warehouse (node F). The arcs of this network represent highway routes with their associated mileage given as the arc lengths.

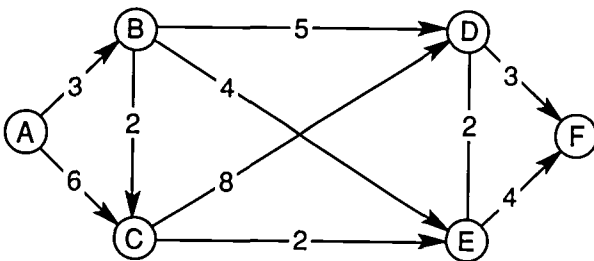


Figure 1. Elementary shipping network.

A commonly encountered problem in such a network is finding the shortest path from the factory to the warehouse. More generally, shortest path calculations are found to be valuable in analyzing the behavior of various large-scale distribution networks, such as complex telecommunications processes, distributed computer architectures, and even lifeline systems subject to seismic risk (ref. 1). The importance of these applications requires that efficient techniques and tools be developed to aid in the network analysis process.

Although techniques for deterministic network analysis are widespread and applicable to large-scale networks (ref. 2), this is not the case for the analysis of stochastic networks, where the behavior of the network components is governed by some random process. The most straightforward approach for investigating the stochastic shortest path problem involves enumeration of all possible states of the network. In general, if the network contains m arcs and each arc can take on k values, there would be k^m states of the network that must be analyzed. This approach is obviously computationally infeasible even for relatively small networks. Thus, current analysis methods have been largely confined to approximation, simulation, and bounding techniques (refs. 3 to 7).

Since the utility of a network model critically depends on the suitability of its assumptions, a general model that does not impose restrictions on the distributional form of the network components is desirable. This report considers a broad class of stochastic networks with the following characteristics: (1) arc values (length, duration, cost, etc.) are discrete random variables, (2) these random variables are statistically independent, and (3) nodes do not fail. These assumptions greatly increase the tractability of the problem while preserving the model's realism by incorporating the random nature of the problem (ref. 8, p. 455). Statistical independence is commonly assumed because it reduces the computational requirements associated with most solution approaches. Additionally, there are no assumptions, such as symmetry of the nodes or arcs, imposed on the architecture of the networks.

This report considers the specific problems of finding the distribution of the length of the shortest or longest (i.e., critical) path through such a stochastic network. The arcs of the network assume random lengths, which could represent, for example, the duration in traversing a communication link or the cost of completing a given function. In real-world settings, these values are most realistically viewed as random variables, rather than as fixed deterministic parameters. Thus, the state of the network depends on the state assumed by each arc. It follows that the shortest (or longest) path through the network is a function of the random arc lengths; hence, path length can be characterized by a probability distribution.

The objective of this report is to present a general algorithm for determining the exact distribution of the shortest path length in a directed, stochastic network. This algorithm has applications to two important classes of stochastic path problems: determining the longest, or critical, path distribution for acyclic networks and the exact two-terminal

reliability for probabilistic networks. The structure of the paper is as follows. In the next section, the necessary notation and basic terminology are defined. The structural decomposition technique is described in the third section, and it is applied to solving the critical path and two-terminal reliability problems in the fourth section. Several network examples are given in this section. A summary of the findings including the limitations of the approach is presented in the last section.

Notation

To facilitate the discussion of probabilistic networks, some basic terminology and notation are first introduced. A probabilistic network is modeled using a directed graph $G = (N, A)$, where N is a set of nodes, representing, for example, warehouses or communication centers in the network, and A is a set of arcs, representing traffic routes or communication buses, connecting certain pairs of nodes. An arc (i, j) , where i and j are elements of N , is defined to be a directed link from the origin node i to the destination node j . If more than one arc connects a pair of nodes, the arcs are denoted with different numbered superscripts. For example, the two arcs spanning nodes D and E in figure 2 are denoted $(D, E)^1$ and $(D, E)^2$. Since we are concerned only with the shortest (or longest) path between two given nodes, we consider only graphs with one source node denoted by s , which has only outgoing arcs, and one terminal node denoted by t , which has only entering arcs. Figure 2 shows a directed graph with source node A and terminal node E. The indegree of node i , $\text{indegree}(i)$, is the number of arcs entering node i , and the outdegree of i , $\text{outdegree}(i)$, is the number of arcs leaving i . For each of the networks considered, $\text{indegree}(s) = 0$ and $\text{outdegree}(t) = 0$. In figure 2, $\text{indegree}(C) = 2$ and $\text{outdegree}(C) = 1$.

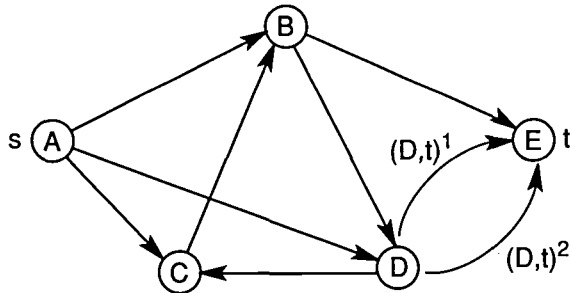


Figure 2. Directed graph.

A path in the graph from node i to node j is defined as an ordered sequence of arcs connecting the two nodes. For example, one possible path from node A to node E in figure 2 is through arcs (A, B) and (B, E) . A cycle is a special type of path

connecting a node to itself. In figure 2, arcs (C, B) , (B, D) , and (D, C) form a cycle. In general, the networks treated here are allowed to contain cycles except in the special case of longest path calculations. In calculating the length of the longest path through a network, the underlying graph must be acyclic since cycles in a graph would lead to a longest path of infinite length.

To incorporate information about the random behavior of the arcs, each arc is assigned a finite n -tuple of nonnegative integer values, indicating, for example, lengths or durations. The n -tuple of arc lengths for arc (i, j) is denoted $a_{(i, j)}$, and the number of lengths associated with arc (i, j) is denoted $|a_{(i, j)}|$. For each arc (i, j) , there is also a corresponding discrete probability distribution for the arc lengths and this is denoted $p_{(i, j)}$. A particular length assumed by arc (i, j) is denoted $L_{(i, j)}$. Suppose the length of arc (B, D) in figure 2 is uniformly distributed on the interval $[6, 9]$. Then the discrete n -tuple of arc lengths is represented by $|a_{(B, D)}| = 4$ and $a_{(B, D)} = \langle 6, 7, 8, 9 \rangle$. The corresponding probability distribution is given by $p_{(B, D)} = \langle 0.25, 0.25, 0.25, 0.25 \rangle$, and $P[L_{(B, D)} = 8] = 0.25$. Note that using discrete probability distributions significantly increases the tractability of the problem in comparison with continuous distributions. Methods for discretizing a continuous distribution, as shown in Dodin (ref. 3), are available when the arc behavior is described by a continuous distribution.

Structural Reduction Techniques

In this section, a graph-theoretic procedure for determining the distribution of the shortest path length through a stochastic network is described. The objective of the approach is to apply certain conditioned reductions to a given network until the network is reduced to an equivalent (with respect to finding the distribution of the shortest path length) network having only two nodes. The arc connecting the two nodes in this reduced network provides the distribution of the shortest path length in the original network.

In this report, two classes of structural reductions are applied to stochastic networks. The first class of reductions represent fundamental simplifications to the topology of the network including the series and parallel reductions. The second class of reductions are based on the concept of local "factoring." The basic difference between the two classes is that the fundamental reductions yield one smaller equivalent network. Local factoring, on the other hand, generates an equivalent set of subnetworks to be solved. Local factoring uses the divide and conquer ideology:

the original problem that is difficult to solve is decomposed into successively smaller problems that are easier to solve. Each type of reduction is discussed more fully in the following subsections.

Fundamental Reductions

The fundamental reductions can be applied to three basic constructs present in a network: series, parallel, and figure-eight. The term *construct* is used throughout this discussion to describe specific configurations of nodes and arcs that exist within a network architecture. Each fundamental reduction simplifies the given network graph by decreasing the number of nodes or the number of arcs (or both).

One of the most easily recognizable constructs within a directed graph is the *series* construct. A series construct exists when some node B in the graph is the intermediate node between exactly two other nodes; that is, when $\text{indegree}(B) = 1$ and $\text{outdegree}(B) = 1$. Figure 3 gives an example of a series construct in which the discrete distribution function on arc (A,B) is uniform on [0,3] and the discrete distribution function on arc (B,C) is uniform on [2,6].

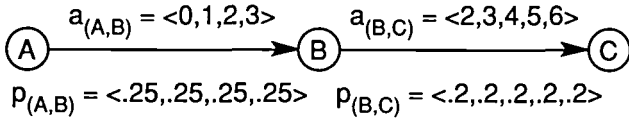


Figure 3. Series construct.

Since any path through node B must include arcs (A,B) and (B,C), these two arcs can be replaced by a single arc between nodes A and C. The set of lengths for the replacement arc (A,C) is the set of all possible additive combinations of arc lengths chosen from $a_{(A,B)}$ and $a_{(B,C)}$. For our example, $L(A,C) = 3$ occurs with a positive probability since the combinations $L(A,B) = 0$ and $L(B,C) = 3$ and $L(A,B) = 1$ and $L(B,C) = 2$ are possible states of that construct. Moreover, because of the statistical independence of arc lengths associated with distinct arcs,

$$\begin{aligned} P[L(A,C) = 3] &= P[L(A,B) = 0]P[L(B,C) = 3] \\ &\quad + P[L(A,B) = 1]P[L(B,C) = 2] \\ &= (0.25)(0.2) + (0.25)(0.2) = 0.1 \end{aligned}$$

More generally, those lengths in the set $a_{(A,C)}$ that occur in several ways are denoted only once in $a_{(A,C)}$ and the corresponding probability is the sum of the contributing probabilities. The reduced construct, which corresponds to the discrete convolution of the arc distributions on (A,B) and (B,C), is shown in figure 4.

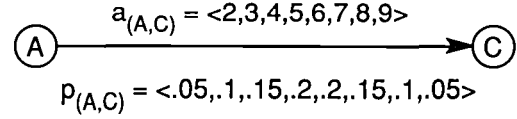


Figure 4. Reduced series construct.

Two or more arcs that connect the same pair of nodes constitute a *parallel* construct. Figure 5 shows a parallel construct where for arc (A,B)¹, $a_{(A,B)^1} = <2,3,5>$ and $p_{(A,B)^1} = <0.25,0.25,0.5>$, the distribution on arc (A,B)² is uniform on [0,3], and the distribution on arc (A,B)³ is uniform on [2,3]. Note that the uniform distribution was used in many examples simply for ease of illustration. In general, the same reduction techniques apply for any discrete distribution.

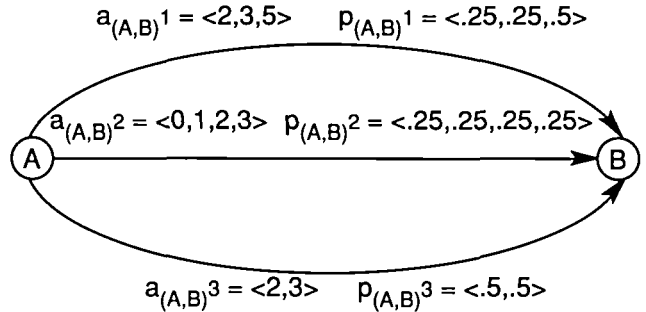


Figure 5. Parallel construct.

The difference between calculating the shortest and longest path lengths through a network manifests itself in reducing the parallel construct. For any given realization of a stochastic network that contains this construct, the shortest or longest path contains at most one of the parallel arcs shown in figure 5. When finding the shortest (longest) path through such a deterministic realization, only an arc having the minimum (maximum) length among these parallel arcs is considered for the shortest (longest) path. It follows that the set of parallel arcs can be replaced by a single arc (A,B), whose arc lengths represent the minimum (maximum) lengths from all possible combinations of arc lengths of the parallel arcs. One possible combination of arc lengths in the example is $L(A,B)^1 = 5$, $L(A,B)^2 = 2$, and $L(A,B)^3 = 3$. When reducing this construct with respect to finding the shortest path length, $\min(5,2,3) = 2$ is included in $a_{(A,B)}$. For this particular combination,

$$\begin{aligned} P[L(A,B)^1 = 5]P[L(A,B)^2 = 2]P[L(A,B)^3 = 3] \\ = 0.0625 \end{aligned}$$

which contributes toward the overall probability $P[L(A,B) = 2]$. Specifically, when considering the

shortest path length for this example,

$$\begin{aligned}
P[L(A,B) = 2] &= \sum_{j, k \geq 2} P[L(A,B)^1 = 2]P[L(A,B)^2 = j]P[L(A,B)^3 = k] \\
&+ \sum_{\substack{j > 2 \\ k \geq 2}} P[L(A,B)^1 = j]P[L(A,B)^2 = 2]P[L(A,B)^3 = k] \\
&+ \sum_{j, k > 2} P[L(A,B)^1 = j]P[L(A,B)^2 = k]P[L(A,B)^3 = 2] \\
&= 0.125 + 0.1875 + 0.09375 = 0.40625
\end{aligned}$$

again using the statistical independence of arc lengths. When considering the longest path length through this example,

$$\begin{aligned}
P[L(A,B) = 2] &= \sum_{j, k \leq 2} P[L(A,B)^1 = 2]P[L(A,B)^2 = j]P[L(A,B)^3 = k] \\
&+ \sum_{\substack{j < 2 \\ k \leq 2}} P[L(A,B)^1 = j]P[L(A,B)^2 = 2]P[L(A,B)^3 = k] \\
&+ \sum_{j, k < 2} P[L(A,B)^1 = j]P[L(A,B)^2 = k]P[L(A,B)^3 = 2] \\
&= 0.09375 + 0 + 0 = 0.09375
\end{aligned}$$

More generally, any length in the set $a_{(A,B)}$ that arises from several combinations is represented only once in $a_{(A,B)}$ and its corresponding probability is the sum of the appropriate individual probabilities. By applying this strategy, the parallel construct in figure 5 can be replaced by one of the single arcs shown in figure 6, depending on whether the shortest or the longest path length is of interest.

The series and parallel reductions are well-known and have applications in many optimization problems. Martin (ref. 9) was the first to apply these reductions to the stochastic shortest path problem. By using these two reduction steps, more complex networks can sometimes be reduced to a single arc between the source and terminal nodes. In fact, a *series-parallel network* is defined to be any network that can be simplified via series and parallel reductions to an equivalent two-node network, that is, equivalent in the sense that the distribution of the shortest (or longest) path length for the original network is exactly the same distribution for the

reduced two-node network. For example, consider the network in figure 7, where the distribution on each arc is uniform on $[1,2]$; that is, $a_{(i,j)} = \langle 1, 2 \rangle$ and $p_{(i,j)} = \langle 0.5, 0.5 \rangle$ for each (i,j) in the graph. By applying only series and parallel reductions, this network can be reduced to the equivalent shortest path network shown in figure 8.

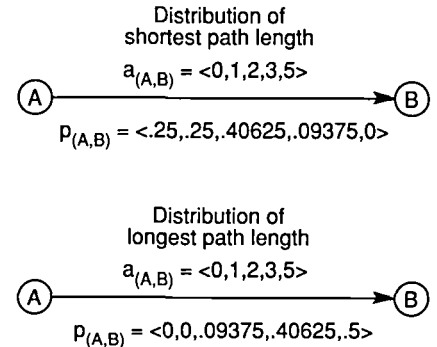


Figure 6. Reduced parallel constructs.

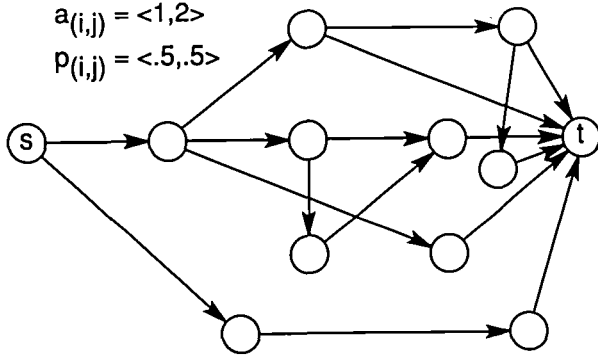


Figure 7. Series-parallel network.

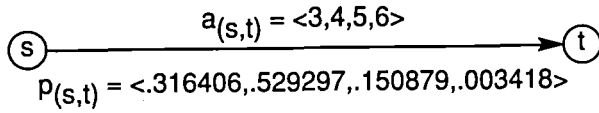


Figure 8. Reduced series-parallel network.

The last fundamental reduction, called a figure-eight reduction, while less well-known than the series and parallel reductions, incorporates the same approach to simplifying a network structure into an equivalent, less complex structure. The *figure-eight* construct, shown in figure 9, is found only in cyclic networks. In this construct, the center node B in the structure must have exactly $\text{indegree}(B) = 2$ and $\text{outdegree}(B) = 2$ and must be connected to exactly two other nodes, labeled A and C in figure 9. Specifically, B has one incoming arc from node A and one from C; both node A and node C have one incoming arc from B.

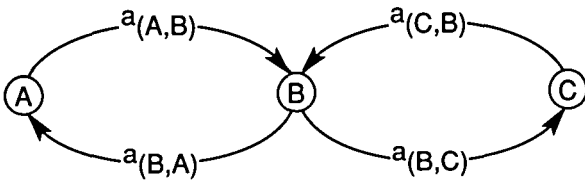


Figure 9. Figure-eight construct.

Since the figure-eight construct occurs only in cyclic networks, only the shortest path from the source node to the terminal node is of interest. In this construct, four possible subpaths exist through node B: $(A,B) \rightarrow (B,A)$, $(B,C) \rightarrow (C,B)$, $(A,B) \rightarrow (B,C)$, and $(C,B) \rightarrow (B,A)$. Any source-to-terminal ($s-t$) path through the network that contains the subpath $(A,B) \rightarrow (B,A)$ must be at least as long as the shortest path through the network. Since this subpath is irrelevant to the distribution of the shortest path length and can be ignored. Similarly, the subpath $(B,C) \rightarrow (C,B)$ can safely be ignored. By eliminating these two paths from the set of possible paths through node B that could be included in

the shortest path, the only remaining paths of interest are $(A,B) \rightarrow (B,C)$ and $(C,B) \rightarrow (B,A)$. The figure-eight construct is then simplified by replacing arcs (A,B) and (B,C) with a single arc (A,C) , and by replacing arcs (C,B) and (B,A) with a single arc (C,A) . The lengths and probabilities of the new arcs are determined in the same manner as with series reductions. For example, the set of lengths for (A,C) is the set of all possible additive combinations of lengths from (A,B) and (B,C) . The simplified figure-eight assumes the form shown in figure 10.

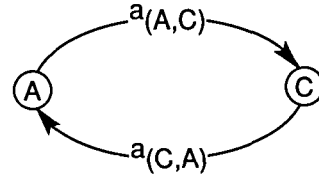


Figure 10. Reduced figure-eight construct.

The advantage of using these fundamental reductions is that they yield a single equivalent network that has a smaller number of arcs and possibly fewer nodes. Hence, determining the distribution of the shortest path length for this reduced network is easier. In fact, these fundamental reductions have been previously applied in calculating network reliability (ref. 10). However, many complex network architectures cannot be simplified via series and parallel reductions. The contribution made to the stochastic shortest path problem through this investigation is the development of a new reduction technique, called conditional factoring, that, like the fundamental reductions, is based on the configuration of the nodes and arcs in the network. Conditional factoring offers additional possibilities for simplifying a given network.

Conditioned Reductions

For a stochastic network whose architecture cannot be simplified by applying series and parallel reductions, most exact methods for finding the distribution of the shortest path length depend on complete state-space enumeration (ref. 2) or enumeration of all cutsets or possible paths (refs. 7, 11, and 12). In simplifying a construct by complete enumeration, all possible combinations of arc lengths for all arcs in that construct are considered. For example, given the network construct in figure 11, if $|[i,j]| = 3$ for each of the 7 arcs in that construct, $3^7 = 2187$ states of that construct would be considered in solving the network. If $|[i,j]| = 10$ for each arc in that construct, $10^7 = 10$ million states of that network are considered. In general, complete enumeration is computationally infeasible except for very small

networks where the arcs can assume only a minimal number of lengths.

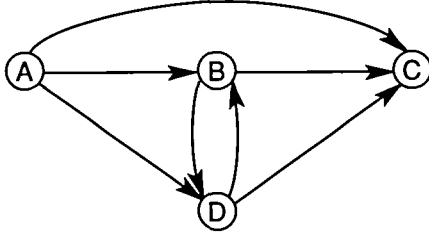


Figure 11. Partial network structure.

In this investigation, some constructs that are prevalent in many network architectures were identified that can be simplified without complete enumeration. The technique used to simplify these constructs, called *conditional factoring*, is based on the idea that it is sufficient to consider only a specific subset of arcs within certain network constructs in order to generate an equivalent, simplified representation. Thus, the conditional reductions can significantly decrease the computational requirements necessary to solve many networks and, hence, allow the exact solution of a larger class of networks. For example, if $|i, j| = 3$ for each arc in figure 11, only 9 states of that construct are needed to solve the network using conditional factoring; and, if $|i, j| = 10$, only 100 states are necessary. In general, conditional factoring can be applied to any node within the network except the source or terminal nodes. However, conditional factoring reduces the computational effort only when it is applied to the four special constructs described in this section. For all other constructs, conditional factoring is believed to be equivalent to complete enumeration.

The concept of conditional factoring is similar to the factoring theorem, which is used for computing network reliability (ref. 13), except that factoring is based on a specific node in the graph instead of an arc. The conditioned reductions are performed by identifying a specific construct centered around a node, called the central node, in the network. The structure of this construct is simplified by removing the central node and completing all possible paths through that central point of that construct. In contrast to the fundamental reductions, conditional factoring generates a set of independent subnetworks to be solved, where each subnetwork incorporates the new, reduced structure into the overall network architecture. These subnetworks are generally smaller in size (i.e., have fewer nodes) and involve less computational effort to solve. The lengths of the new arcs in the subnetworks are defined by conditioning on the arcs that are used more than once in defining new arcs in the reduced construct. These arcs that are

used more than once in forming the new arcs of the reduced construct are called *factoring arcs*. A unique subnetwork is defined for each combination of lengths among the factoring arcs, and this eliminates any dependencies among the subnetworks. In this section, the concept of conditional factoring is illustrated by applying it to four special network constructs.

The first construct to be introduced is the *fan* construct. A *deterministic fan* construct with central node B is shown in figure 12. Note that in any deterministic construct, the length of each arc is known with certainty; that is, the arc values are not random variables. In general, a fan construct with central node B has the following characteristics: (1) B is any node in the network except the source or terminal node, and (2) either $\text{indegree}(B) = 1$ or $\text{outdegree}(B) = 1$ (if both are 1, then B is the central node in a series structure). If $\text{indegree}(B) = 1$, the incoming arc to node B is called the factoring arc (since it is the only arc in the construct used more than one time in reducing the construct) and is denoted $f(B)$. Similarly, if $\text{outdegree}(B) = 1$, the outgoing arc from node B is called the factoring arc and is denoted $f(B)$. In figure 12, the factoring arc is $f(B) = (A, B)$.

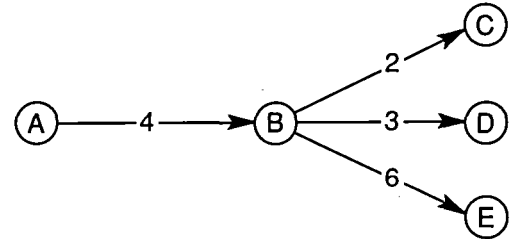


Figure 12. Deterministic fan construct.

In a deterministic construct, each arc (i, j) has a constant length, say $k(i, j)$. Thus, within that construct, $|i, j| = 1$ and $P[L(i, j) = k(i, j)] = 1$. If two arcs, (A, B) and (B, C) , with constant lengths are combined, the length of the resulting arc is the sum of the two arc lengths. Thus, the new length $L(A, C) = k(A, B) + k(B, C)$ is also a constant and has corresponding probability of 1.

A fan construct is graphically simplified by eliminating the central node and completing each possible path (relative to the graph) through that central connection point. In a deterministic fan construct, the length of the factoring arc is added to the lengths of each of the other arcs connected to the central node. For example, arc (A, C) in figure 13 is formed from arcs (A, B) and (B, C) , arc (A, D) is formed from arcs (A, B) and (B, D) , and arc (A, E) is formed from arcs (A, B) and (B, E) . The length of each new arc is accordingly the sum of lengths of its component arcs; and, since each arc has a constant length, the

probability associated with each new arc is 1. For the network in figure 12, this reduction process yields the network in figure 13.

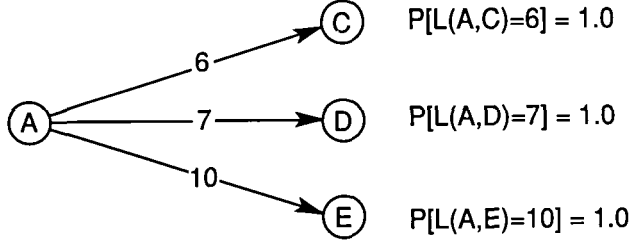


Figure 13. Reduced deterministic fan construct.

The concept of conditional factoring applies when the length of each arc is assumed to be a random variable. To illustrate, consider the same graph shown in figure 12 but allow each arc to be a random variable. Figure 14 shows this construct where each arc can take on values in the n -tuple $\langle 1, 2 \rangle$ with corresponding probabilities $\langle 0.4, 0.6 \rangle$. Such a *stochastic fan* construct centered at node B is decomposed based on the distribution of lengths on the factoring arc $f(B)$. In figure 14, $f(B) = (A,B)$ just as in figure 12.

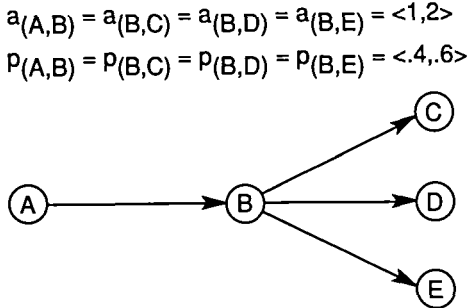


Figure 14. Stochastic fan construct.

Since the length of $f(B)$ is now a random variable, a distinct subnetwork is generated for each length in $a_{f(B)}$. The number of subnetworks created by factoring on node B is denoted $\omega(B)$. For the stochastic fan construct, the number of subnetworks generated is equal to the number of possible lengths for the single factoring arc; that is, $\omega(B) = |f(B)|$. To generate each subnetwork S_j , for $j = 1, 2, \dots, \omega(B)$, the following steps are taken: (1) let $f(B)$ take on a constant length c_j , where c_j is the j th length in $a_{f(B)}$; (2) eliminate node B from the original construct; (3) complete all possible paths through the central point; and (4) define the new arc lengths and the associated probabilities.

In general, the arc lengths in the reduced construct are determined by adding the lengths of their component arcs. In the strictly deterministic construct, this is simple since only constant length arcs are combined. However, in reducing a stochastic construct, constant length arcs (resulting from the

factoring arc) are combined with arcs with random lengths. When an arc with a constant length is combined with an arc whose length is governed by a discrete distribution D , the length of the resulting arc is a random variable governed by D .

The fan construct shown in figure 14 can be decomposed into an equivalent set of two independent subnetworks, shown in figure 15, since $|f(B)| = 2$. The first subnetwork S_1 is constructed by assuming $L(A,B) = 1$, and the second subnetwork S_2 is constructed for $L(A,B) = 2$. The probability of obtaining subnetwork 1 is $P(S_1) = P[L(A,B) = 1] = 0.4$, and, similarly, the probability of obtaining subnetwork 2 is $P(S_2) = P[L(A,B) = 2] = 0.6$.

The next three reductions focus on constructs with a central node B where either $\text{indegree}(B) = 2$ or $\text{outdegree}(B) = 2$. Additionally, each of these constructs contains at least one simple cycle of size two (a cycle involving only two nodes) involving the central node B.

The first of these constructs, shown in figure 16, is called the *loop* construct with central node B. In general, a loop construct with central node B has the following characteristics: (1) $\text{indegree}(B) = 2$, (2) $\text{outdegree}(B) = 2$, and (3) B and some other node D form the only cycle of the construct, which is a simple cycle. The possible paths through this structure that are considered for the shortest path are then $(A,B) \rightarrow (B,D)$, $(A,B) \rightarrow (B,C)$, and $(D,B) \rightarrow (B,C)$. The only other path through the structure, $(A,B) \rightarrow (B,D) \rightarrow (D,B) \rightarrow (B,C)$ is never considered as part of a shortest path since a subset of that path, $(A,B) \rightarrow (B,C)$, is always at least as short. As a result, we can replace each of these three paths with an equivalent arc, as shown by the reduced construct shown in figure 17.

Since the arcs (A,B) and (B,C) are used more than once in defining the new arcs in the reduced construct, the lengths of both of these arcs must be considered when generating subnetworks. Hence, arcs (A,B) and (B,C) are factoring arcs for this structure and are denoted $f_1(B)$ and $f_2(B)$, respectively. In general, the number of subnetworks produced by any conditioned reduction is the product of the distribution sizes for each factoring arc. For a loop construct, a distinct subnetwork is produced for each combination of lengths for the two factoring arcs, so $\omega(B) = |f_1(B)| |f_2(B)|$. To generate the subnetwork S_k , where $k = 1, 2, \dots, \omega(B)$, the following steps are taken: (1) let the factoring arcs $f_1(B)$ and $f_2(B)$ take on one of the combinations of constant values from $a_{f_1}(B)$ and $a_{f_2}(B)$; (2) eliminate node B from the original structure; (3) complete all possible paths through the central point; and (4) appropriately define the new arc lengths and their probabilities.

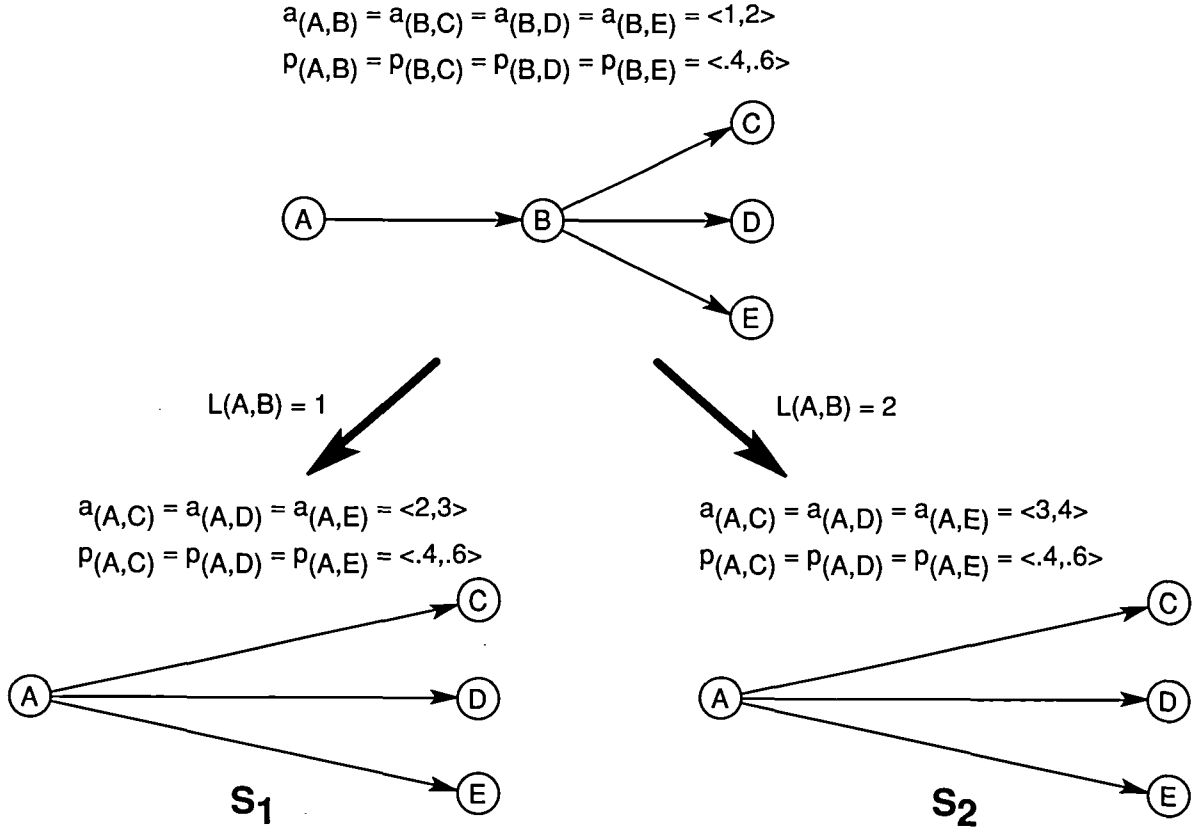


Figure 15. Subnetworks for the stochastic fan construct.

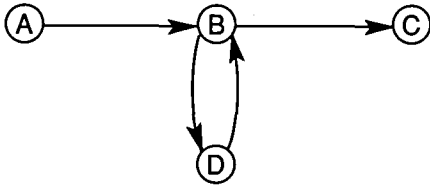


Figure 16. Loop construct.

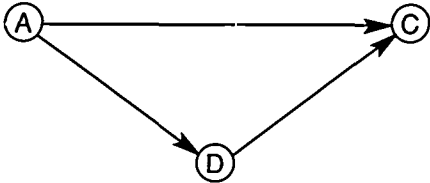


Figure 17. Reduced loop construct.

To illustrate, suppose that in the loop construct of figure 16 each arc can take on lengths in the n -tuple $\langle 1, 2 \rangle$ with equal probability. From the first reduction step, four distinct subnetworks are generated where $L(A,B) = 1$ and $L(B,C) = 1$ are in S_1 , $L(A,B) = 1$ and $L(B,C) = 2$ are in S_2 , $L(A,B) = 2$ and $L(B,C) = 1$ are in S_3 , and $L(A,B) = 2$ and

$L(B,C) = 2$ are in S_4 . These four subnetworks are shown in figure 18. To simplify the notation in the following figures, the lengths and probabilities are displayed in an abbreviated form. In this notation, the lengths and corresponding probabilities for each arc appear in the format: $\langle \text{length}, \text{probability} \rangle$.

The third construct is a *general loop* construct and is shown in figure 19. The general loop construct with central node B has the following characteristics: (1) either $\text{indegree}(B) = 2$ and $\text{outdegree}(B) > 2$ or $\text{outdegree}(B) = 2$ and $\text{indegree}(B) > 2$; and (2) B together with some node A forms the only cycle of the construct, which again is a simple cycle. When the central node is removed from a general loop construct, the number of arcs in each resulting subnetwork is at least as great as the number of arcs in the original network. For a general loop construct with central node B, if $\text{indegree}(B) + \text{outdegree}(B) = 5$, the number of arcs in the reduced construct is the same as in the original construct. If $\text{indegree}(B) + \text{outdegree}(B) > 5$, the reduced construct contains more arcs than the original.

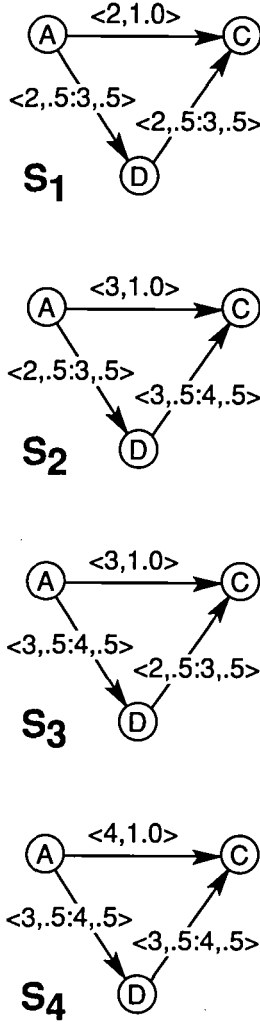


Figure 18. Subnetworks for the reduced loop construct.

The reduction of a general loop construct is illustrated in figure 20. In this case, the reduced construct has one more arc than the original construct. Note that in reducing the general loop construct each arc in the original structure is used more than once except for arc (B,A). As a result of this commonality, each arc in the original general loop construct must be a factoring arc except for arc (B,A). The number of factoring arcs γ is thus $\gamma = \text{indegree}(B) + \text{outdegree}(B) - 1$ and the number of subnetworks generated is $\omega(B) = |f_1(B)| |f_2(B)| \dots |f_\gamma(B)|$. As in the fan and loop constructs, the basic steps to generate subnetwork S_k , where $k = 1, 2, \dots, \omega(B)$, are as follows: (1) let the factoring arcs $f_i(B)$, where $i = 1, 2, \dots, \gamma$, take on one of the combinations of constant lengths in $a_{f_i}(B)$; (2) eliminate node B from the original structure; (3) complete all possible paths through the central point; and (4) appropriately define the new arc lengths and probabilities.

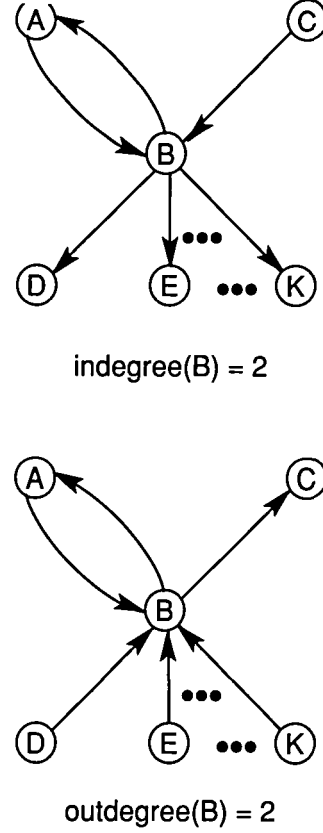


Figure 19. General loop constructs.

Suppose each arc in the general loop structure of figure 20 can assume an integer length in the n -tuple $\langle 1, 2 \rangle$ with equal probability. Then, there would be $\gamma = 5$ factoring arcs, and $\omega(B) = 2^5 = 32$ subnetworks generated in reducing that structure, compared with $2^6 = 64$ networks that would be generated using complete enumeration. Two of the subnetworks are shown in figure 21. In these subnetworks, only arc (C,A) has a nonconstant distribution. This is because (C,A) is formed by combining arcs (B,A) and (C,B), and arc (B,A) is the only arc in the structure that is not a factoring arc of node B.

The last conditioned reduction construct, shown in figure 22, is called a *double loop* construct. Like the general loop construct, either the indegree or the outdegree of the central node must be 2. The double loop construct, though, must contain exactly two simple cycles connected to the central node. In general, a double loop construct with central node B has the following characteristics: (1) either $\text{indegree}(B) = 2$ and $\text{outdegree}(B) > 2$, or $\text{outdegree}(B) = 2$ and $\text{indegree}(B) > 2$; and (2) B forms exactly two simple cycles: one with some node A and one with some other node C; see figure 22.

When the double loop construct is reduced, two of the arcs in the original network are used only one time, such as arcs (B,A) and (B,C) for the construct with $\text{indegree}(B) = 2$ and arcs (A,B) and (C,B) for the construct with $\text{outdegree}(B) = 2$ in figure 22. As with all conditioned reductions, each arc that is used more than once in creating the reduced construct is a factoring arc. Hence, for any double loop construct with central node B there are $\gamma = \text{indegree}(B) + \text{outdegree}(B) - 2$ factoring arcs, and the total number of subnetworks is $\omega(B) = |f_1(B)||f_2(B)| \dots |f_\gamma(B)|$. The steps used to generate each subnetwork of a double loop construct are the same as those used for the other conditioned reductions. An example of the graphical reduction of a double loop construct is shown in figure 23. As with the general loop construct, removing the central node of a double loop construct can produce more arcs in each resulting subnetwork than are present in the original construct. When $\text{indegree}(B) + \text{outdegree}(B) > 6$, more arcs are present in the reduced construct than in the original. Some of these additional arcs, though, may lead to further parallel reductions.

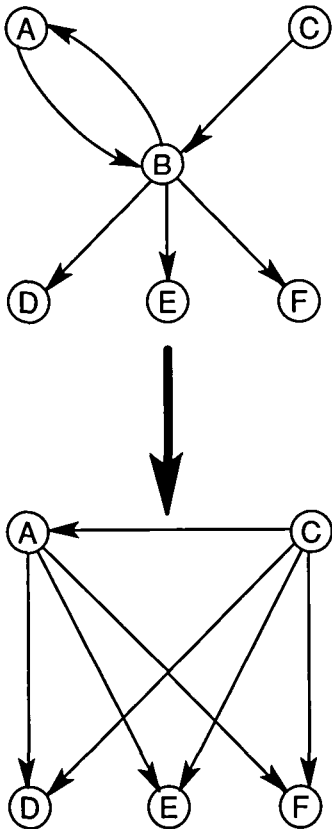


Figure 20. Reduction of general loop construct.

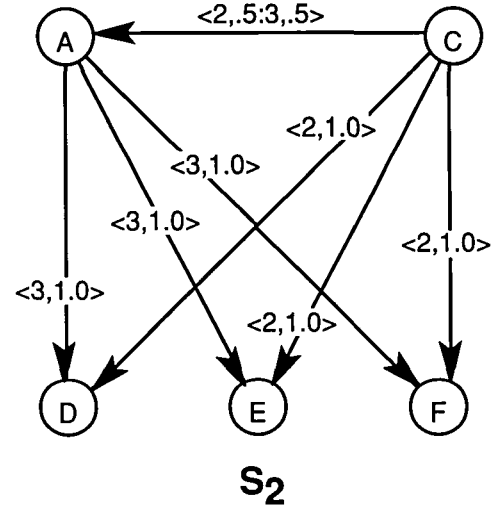
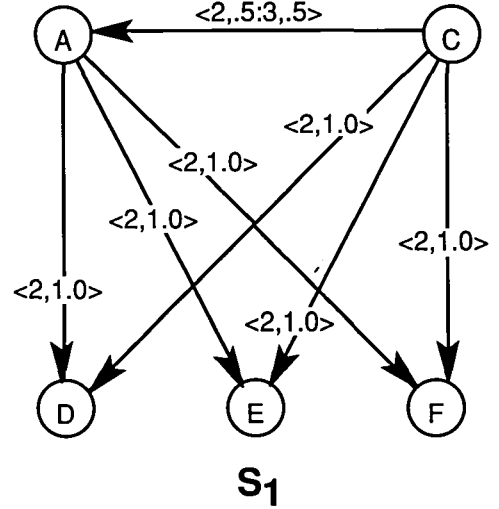
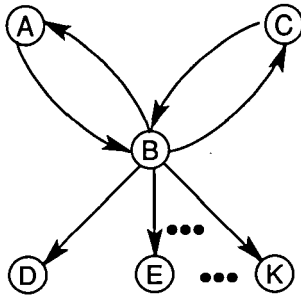


Figure 21. Two subnetworks for the reduced general loop construct.

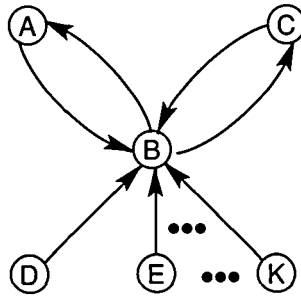
Conditioned Reductions and Shortest Path Calculations

The rules for simplifying a special construct present within a stochastic network via conditional factoring follow a general scheme. Namely, once one of the constructs discussed in the preceding section has been identified, the following steps are taken to generate the desired subnetworks:

1. Identify all factoring arcs for that construct.
2. List all possible combinations of lengths for the factoring arcs. Each combination defines a distinct subnetwork.
3. For each combination of lengths, compute the product of the probabilities associated with each length in the combination. Because of



$\text{indegree}(B) = 2$



$\text{outdegree}(B) = 2$

Figure 22. Double loop constructs.

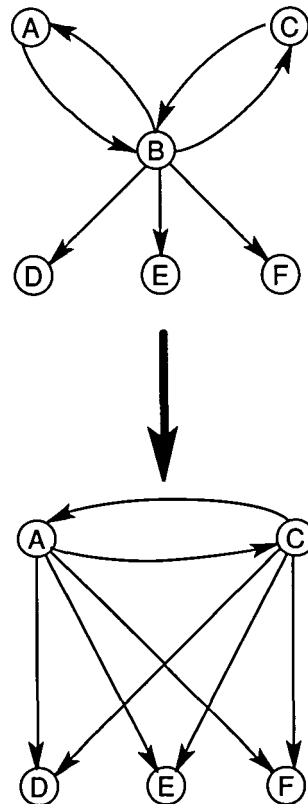


Figure 23. Reduction of a double loop construct.

the statistical independence assumption, this product is the probability of that subnetwork occurring.

4. Remove the central node from the construct and graphically complete all possible paths through the construct. (Note that any loop arc (i, i) formed can be eliminated from that subnetwork structure.)
5. Add the lengths of the component arcs to get the lengths of the new arcs in the subnetworks.
6. Define the probabilities for the new arcs. An arc with a constant length has probability 1. When an arc with a constant length is combined with an arc whose length is governed by a discrete distribution D , the length of the resulting arc is a random variable governed by D .

By following these steps, a network can be decomposed into a set of subnetworks that are more easily solved. To demonstrate how these steps are used to calculate the exact distribution of the shortest path length, consider the network shown in figure 24; notice that this network is not a series-parallel network. Suppose each arc in figure 24 can take on lengths in the n -tuple $\langle 1, 3 \rangle$ with equal probability.

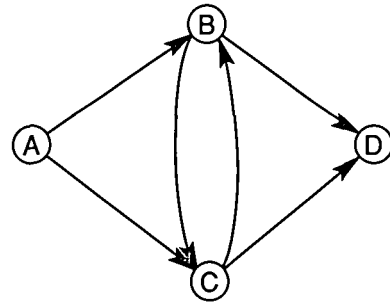


Figure 24. Network with loop construct.

In this network, there is a loop construct with central node B and a loop construct with central node C. For this example, the choice of central node is not important, so node B is arbitrarily selected. The heuristic rules used to govern the choice of which node to remove are discussed in a later section. According to steps 1 and 2 of the decomposition steps, there will be four subnetworks generated from the four possible combinations of arc lengths from the factoring arcs (A, B) and (B, D) . These combinations are $L(A, B) = 1$ and $L(B, D) = 1$ in S_1 , $L(A, B) = 1$ and $L(B, D) = 3$ in S_2 , $L(A, B) = 3$ and $L(B, D) = 1$ in S_3 , and $L(A, B) = 3$ and $L(B, D) = 3$ in S_4 . The four subnetworks are shown in figure 25. From step 3 of the decomposition steps, the probability

associated with each subnetwork is the product of the probabilities corresponding to the constant lengths used to define that subnetwork. In figure 25,

$$P(S_1) = P[L(A, B) = 1]P[L(B, D) = 1] = 0.25$$

$$P(S_2) = P[L(A, B) = 1]P[L(B, D) = 3] = 0.25$$

$$P(S_3) = P[L(A, B) = 3]P[L(B, D) = 1] = 0.25$$

$$P(S_4) = P[L(A, B) = 3]P[L(B, D) = 3] = 0.25$$

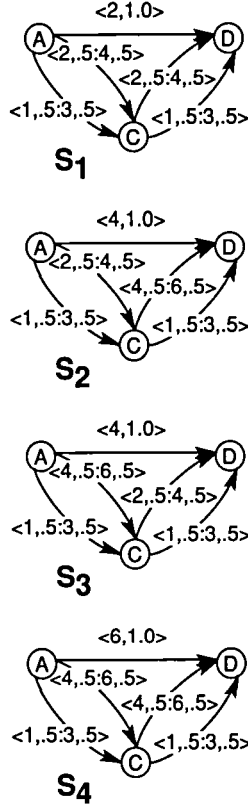


Figure 25. Subnetwork structures for the reduced network.

Notice that each subnetwork in figure 25 can now be simplified to an equivalent two-node network, with respect to finding the distribution of the shortest path length, via series and parallel reductions. These simplified networks are shown in figure 26.



Figure 26. Reduced subnetworks.

The distribution of the shortest path length through the original network in figure 24 is calculated as shown in table 1. The distribution for the original network is found by combining the shortest path distributions in each subnetwork S_i using as weights the appropriate $P(S_i)$; that is, for each subnetwork, the probability of each path length is multiplied by the probability of that subnetwork occurring and the resulting probabilities are summed across all subnetworks. Note that the columns for the distributions in table 1 all sum to 1.0 since the shortest path length distribution for each individual subnetwork must sum to 1.0.

Table 1. Distribution of Shortest Path Length in the Network With a Loop Construct

Length of shortest path	Probability in final subnetworks				Probability for original network
	S_1 $P(S_1) = 0.25$	S_2 $P(S_2) = 0.25$	S_3 $P(S_3) = 0.25$	S_4 $P(S_4) = 0.25$	
2	1.0	0.25	0.25	0.25	0.4375
3	0	.125	.125	0	.0625
4	0	.625	.625	.5	.4375
6	0	0	0	.25	.0625

Total Factoring

Although the special constructs required for the basic and conditioned reductions are found in many network configurations, realistic network configurations often do not, on initial inspection, contain any of these constructs. In these cases, the factoring approach and the steps for conditioned reductions can still be applied; however, all arcs that are incident with the node chosen for removal are now factoring arcs. That is, all possible combinations of arc lengths around a central node are completely enumerated. If total factoring is required for each node of the network except for the source and terminal nodes, this approach is equivalent to complete enumeration, which, as discussed earlier, is computationally infeasible for all but the smallest networks. Fortunately, applying total factoring on a node in a complex network often yields subnetworks that can be simplified via the basic and conditioned reductions.

For example, the network shown in figure 27 does not contain any of the special constructs discussed in the previous sections.

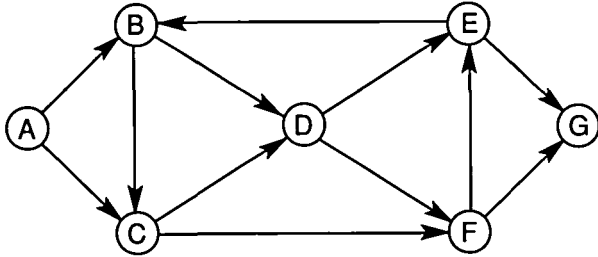


Figure 27. Network with no special constructs.

To demonstrate total factoring, let node B be the central node to be removed from the network. The subnetworks generated by removing node B each have the form shown in figure 28.

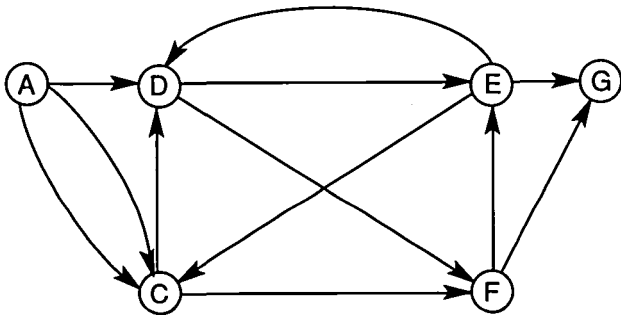


Figure 28. Subnetwork structure after total factoring.

After performing a parallel reduction on the arcs connecting nodes A and C, we can now identify a general loop construct with central node D in this subnetwork structure. When node D is removed, the new subnetwork shown in figure 29 results. In this

new subnetwork, there is a simple loop construct with central node C, a general loop construct with central node F, and a double loop construct with central node E. Since there are fewer factoring arcs associated with the simple loop construct than with the general or double loop constructs, node C is chosen for factoring. After factoring on node C, the subnetworks whose structure is shown in figure 30 are generated. After performing all appropriate parallel reductions, these subnetworks have the same graphical form as the example given in figure 24 and, thus, can be solved in the same manner.

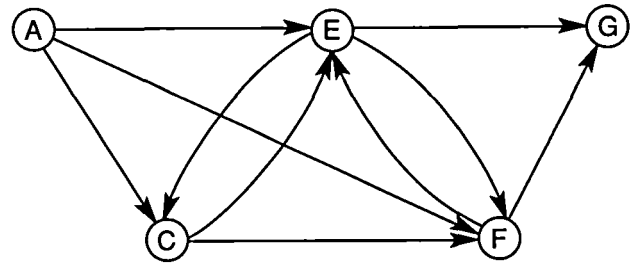


Figure 29. Second subnetwork structure after reduction of general loop construct.

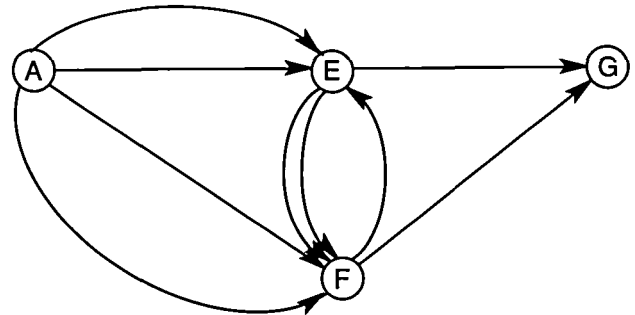


Figure 30. Third subnetwork structure after reduction of simple loop construct.

Determining the Shortest Path Length Distribution

In previous sections, the distribution of the shortest path length has been found for series-parallel networks and for networks where only one level of subnetworks needs to be generated. For many networks, such as the one given in figure 27, reductions must be repeatedly applied before the original network is simplified to an equivalent two-node network. That is, each generated subnetwork can itself be reduced by further application of conditioned and fundamental reductions.

In general, the final subnetworks that simplify to two-node structures (weighted by their associated probabilities of occurrence) are the only contributors to the overall distribution of the shortest path length

in the network. Hence, for each possible subnetwork, it is important to keep track of the probability of that subnetwork occurring. Suppose a conditioned reduction is applied to a network G , and several subnetworks, denoted S_k , are generated. Then, by the statistical independence assumption, the probability of each subnetwork S_k is calculated according to step 3 in the general reduction procedure. Now suppose that another conditioned reduction is used to simplify subnetwork S_k and additional j subnetworks, denoted $S_{k,j}$, are generated. The current status of the computations can be represented by the tree depicted in figure 31. The shaded leaves of this subnetwork tree represent the currently unresolved subnetworks.

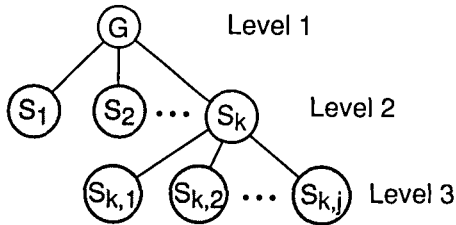


Figure 31. Subnetwork tree.

The probability of the j th subnetwork of S_k occurring is $P[S_{k,j}] = P[S_{k,j}|S_k]P[S_k]$, where $P[S_{k,j}|S_k]$ is calculated according to step 3 in the decomposition procedure. This process of generating new subnetworks from existing subnetworks continues until a generated subnetwork simplifies to a two-node structure.

Note that all subnetworks at level 2 of the subnetwork tree have the same graphical structure. The only differences among these subnetworks are the lengths and probabilities occurring on the arcs. More generally, all subnetworks on the same level of the subnetwork tree have the same graphical structure. So if a conditioned reduction is required to simplify S_k , the same conditioned reduction applies to S_1, S_2, \dots, S_{k-1} . Also, if some $S_{k,j}$ in figure 31 simplifies to a two-node network without additional conditioned reductions, each subnetwork on level 3 also simplifies to a two-node network without generating further subnetworks. These characteristics of the subnetwork tree suggest that the reduction algorithm may be amenable to parallel implementation.

To calculate the distribution of the shortest path length in the original network, only the subnetworks at the final level of the subnetwork tree are considered; that is, those leaf subnetworks that simplify to two-node networks without further conditioned reductions. For each leaf subnetwork in the final level, the probability of each arc length appearing in the associated distribution is multiplied by the proba-

bility of that subnetwork occurring. Then, for every possible length, the contributing probabilities are summed across all leaf subnetworks. This yields the distribution of the shortest path length in the original network.

This process of calculating the distribution of the shortest path length has been implemented in a computer program. The program is written in FORTRAN 77 and is implemented on a Digital Equipment Corporation VAXstation 3200 workstation. The algorithm for determining the distribution of the shortest path length is given as follows:

```

Input network data and store it
Use all applicable basic reductions to simplify the
network
If the network has been completely reduced
then
    Store the distribution information (note:
    this is the complete distribution of the
    shortest path length)
else
    Identify a node to factor on
    Apply the appropriate reduction, generate
    the subnetworks, and place them in a
    stack
    While there are subnetworks in the stack
        Remove the subnetwork on the top of
        the stack
        Use basic reductions to simplify the
        subnetwork
        If the subnetwork has been completely
        reduced
            then
                Store the distribution informa-
                tion for that subnetwork
            else
                Identify a node within the sub-
                network to factor on
                Apply the appropriate reduc-
                tion, generate new subnetworks
                and place them on top
                of the stack
            end if condition
        end while condition
    end if condition
Print the distribution of the shortest path length.

```

The algorithm used to find the distribution of the longest path length is identical except that maximum in place of minimum values are used in computing the parallel reductions.

In theory, this algorithm is valid for an arbitrarily large network. However, the implementation of the algorithm is limited since the number of subnetworks that must be generated and stored to solve a problem can overwhelm the computational resources available. In general, the size of the subnetwork tree depends on the complexity, number of nodes, and size of the distribution of lengths for each arc in the original network. If there are n nodes in the original network, there will be at most $n - 1$ levels in the subnetwork tree. If there are k subnetworks generated during each simplification, then there would be a total of $1 + k + k^2 + \dots + k^{n-2} = (k^{n-1} - 1)/(k - 1)$ subnetworks to solve. However, the maximum number of subnetworks that ever need to be stored is $(k-1)(n-1) - (k-2)$. These are worst-case estimates.

In implementing the algorithm, an effort was made to minimize the size of the subnetwork tree by judiciously choosing the node for factoring. Specifically, the factoring node i is chosen to fulfill the following criteria:

1. For all nodes j in the current subnetwork, $\min[\text{indegree}(i), \text{outdegree}(i)] \leq \min[\text{indegree}(j), \text{outdegree}(j)]$.
2. Among all nodes k that satisfy condition 1, $\omega(i) \leq \omega(k)$.
3. Among all nodes that satisfy conditions 1 and 2, i is the first node that occurs in the graph.

Empirical evidence has shown that in many cases, choosing the factoring node according to these criteria results in a smaller total number of subnetworks that need to be solved. However, these criteria are not optimal for all networks. Further research is necessary to determine whether globally optimal criteria exist for choosing the order of factoring. In practice, the number of subnetworks solved and the maximum number stored have been far fewer than those predicted by the worst-case estimates. The results presented in the next section demonstrate that a fairly modest computational effort is usually required.

Applications of the Structural Factoring Approach

In this section, the structural factoring approach is applied to several network problems taken from the literature. Two of the more interesting problems associated with network analysis are determining the distribution of the longest, or critical, path and the

two-terminal reliability of a network. These two problems are actually special cases of the stochastic shortest path problem.

Critical Path Analysis

Although the analysis of the critical path is limited to acyclic networks, critical path analysis has many applications to scheduling and performance problems such as those associated with communications networks. To determine the distribution of the length of the critical path, the structural reductions are applied to an acyclic network just as they are in determining the shortest path except for decomposing a parallel construct. Recall that in reducing a parallel construct, the maximum arc lengths are considered instead of the minimum arc lengths. For example, consider the network problem presented by Fulkerson (ref. 6). The network has a simple four-node architecture, shown in figure 32, where the length of each arc is uniformly distributed on the interval $[0,2]$.

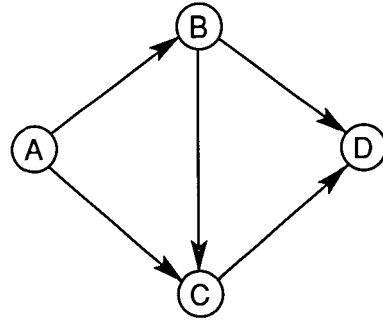


Figure 32. Fulkerson network (ref. 6).

In this network, two fan constructs can be identified, one centered at node B, since $\text{indegree}(B) = 1$, and the other centered at node C, since $\text{outdegree}(C) = 1$. The distribution of the longest path length through the network has been determined using the implementation of the algorithm and is given in table 2. From the distribution in table 2, the expected length of the critical path is calculated to be 3.32510288. In the Fulkerson paper, the lower bound calculated by Fulkerson's method is given as 3.22. This problem required 3.5 seconds of central processing unit (cpu) time to solve for the exact distribution and four subnetworks were generated.

Another renowned problem, shown in figure 33, is the crossing network, also referred to as the "wheatstone bridge" network, analyzed by Kleindorfer (ref. 14). Since this network is acyclic, the distributions for the length of the shortest and longest paths can be calculated. In his paper, Kleindorfer gives bounds on the cumulative distribution for this network where each arc assumes a length in the set

$\langle 1, 2, 3, 4, 5 \rangle$ with equal probability. By applying structural factoring to the fan structures in this network, the exact distribution of the shortest path length from node A to node F can be determined. Table 3 gives the the distribution of the shortest path for this network; table 4 shows the cumulative distribution of the critical (longest) path through the same network as well as the Kleindorfer bounds.

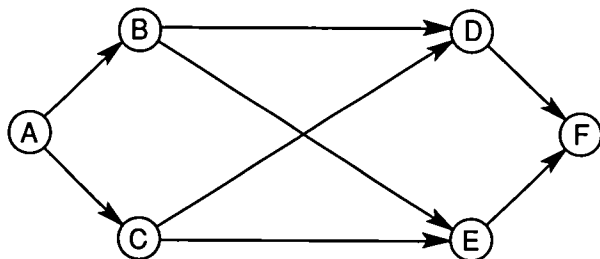


Figure 33. Kleindorfer crossing network (ref. 14).

Table 2. Distribution of Longest Path Length for the Fulkerson Network

Length of longest path	Exact probability
0	1/243
1	11/243
2	49/243
3	74/243
4	72/243
5	27/243
6	9/243

Table 3. Distribution of Shortest Path Length for the Crossing Network

Length of shortest path	Probability
3	0.03064064
4	.08365312
5	.14335488
6	.18986496
7	.20426496
8	.16326144
9	.10479360
10	.05362176
11	.02052864
12	.00505344
13	.00087552
14	.00008448
15	.00000256

Table 4. Distribution of Critical Path Length for the Crossing Network Compared With the Kleindorfer Cumulative Bounds

Length of critical path	Kleindorfer lower bound (ref. 14)	Cumulative probability	Kleindorfer upper bound (ref. 14)
3	0	0.00000256	0.008
4	0	.00008704	.032
5	0	.00096256	.080
6	0.002	.00601600	.160
7	.014	.02654464	.280
8	.055	.08016640	.424
9	.149	.18496000	.576
10	.312	.34822144	.720
11	.528	.55248640	.840
12	.731	.74235136	.920
13	.882	.88570624	.968
14	.969	.96935936	.992
15	1.000	1.00000000	1.000

For critical path problems, the expected project duration time, which is the expected length of the critical path, is often of interest. The expected project duration was calculated from the exact distribution of the critical path length to be 11.203136. Kleindorfer gives a lower bound (LB) of 9.000 and an upper bound (UB) of 11.358 for this problem. Shogan (ref. 15) also gives bounds on the expected project duration for this problem: LB = 10.6 and UB = 11.358. Only 3.7 seconds of cpu time were required to solve this problem exactly.

Since each arc in this network has five possible lengths, the complete enumeration approach would consider $5^8 = 390\,625$ possible states of the network to determine exactly the distribution of the shortest path length. For each possible state of the network, the shortest (or longest) path length must be identified. In contrast, to solve this problem, the structural factoring approach generated only 31 subnetworks (of which 30 were less complex than the original network). A computer program that implements the complete enumeration approach took 57.21 seconds of cpu time to execute, and the resulting distribution agreed with the one given in table 3. The conditional factoring approach is clearly a drastic improvement over complete enumeration for determining an exact solution.

Two-Terminal Reliability

The translation of the stochastic shortest path problem to the two-terminal reliability problem is not

as obvious. The two-terminal reliability problem is defined relative to a network G in which each arc (i, j) has a given probability $p(i, j)$ of functioning independent of all other arcs. The two-terminal reliability of the network is equal to the probability that there exists at least one path in G from a specified node s to a specified node t along which all arcs are functioning.

Now consider a stochastic network G' involving the same nodes and same arcs as G . Each arc (i, j) in G' can assume only the lengths 0 and 1, with probabilities $p(i, j)$ and $1 - p(i, j)$, respectively. If there is a shortest s - t (source to terminal node) path through G' that has length 0, then there must be some path in G' on which all arc lengths are 0. This directly corresponds to a path in G composed of all functioning arcs. Similarly, if the network is functioning, then there must be a path through the G' network that has length 0, and this will be a shortest s - t path. Thus, the probability that the shortest s - t path through G' has length zero is precisely the two-terminal reliability of the original network G . There are no architectural restrictions on the network when computing the two-terminal reliability except that there be a source and terminal node. Thus, the exact two-terminal reliability can be determined for cyclic as well as acyclic networks. Network reliability problems are known however to be #P-complete (ref. 1) where #P is a class of counting problems analogous to NP. Hence, #P-complete problems are at least as difficult to solve, if not more difficult than NP-complete problems such as the traveling salesman problem (ref. 1).

The next example is a cyclic network, shown in figure 34, presented by Shogan (ref. 16). Two loop constructs, at nodes B and D, can be identified in the original network structure.

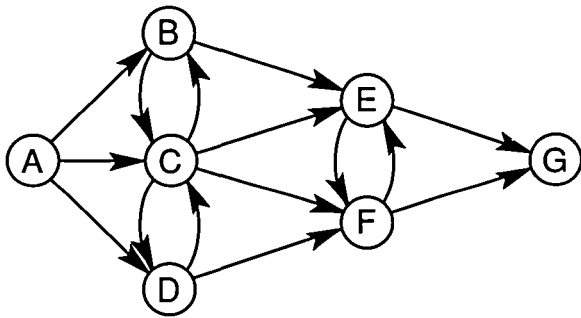


Figure 34. Shogan network (ref. 16).

For this example, the n -tuple of lengths for each arc is $\langle 0, 2, 4 \rangle$ with corresponding probabilities $\langle 0.4, 0.2, 0.4 \rangle$. The exact distribution of the shortest path length through this network is given in table 5.

Table 5. Distribution of Shortest Path Length for the Shogan Network

Length of shortest path	Probability
0	0.3047540654
2	.2810753843
4	.2784418939
6	.0986955899
8	.0339240387
10	.0028468838
12	.0002621440

To solve this network for the distribution of the shortest path length, 1528 subnetworks were generated, but the maximum number of networks stored by the computer at any given time was 27. Although a large number of subnetworks were generated, this problem took only 8.19 seconds of cpu time to solve. By contrast, it took 6048.89 seconds of cpu time to solve the same problem by completely enumerating all $2^{15} = 32768$ possible combinations of path lengths.

This problem can also be considered as a two-terminal reliability problem where the probability that each arc is operational is 0.4. The two-terminal reliability of the network, as shown in table 5, is 0.3047540654. Shogan gives the following bounds on the two-terminal reliability: LB = 0.1971 and UB = 0.3527.

The last example, shown in figure 35, is a large network with 20 nodes and 38 arcs that was also presented by Kleindorfer (ref. 14). The two-terminal reliability of this network, in which each arc is operational with probability 0.9, can be determined by letting each arc take on lengths $\langle 0, 1 \rangle$ with corresponding probabilities $\langle 0.9, 0.1 \rangle$. Using the program with these parameters, the s - t reliability of this network is found to be 0.98612801. The subnetwork tree generated for this problem had 15 levels. A maximum of 15 subnetworks were stored at any given time, and 32767 total subnetworks were generated. The program required 49.21 seconds of cpu time to solve this problem. For this problem, complete enumeration would not be feasible since 2^{38} or approximately 2.75×10^{11} states of the network would be individually examined to find the shortest path length through each. Based on the time to solve a portion of this network by complete enumeration, approximately 280 000 000 seconds of cpu time (or approximately 9 years) would be required to solve the large Kleindorfer network.

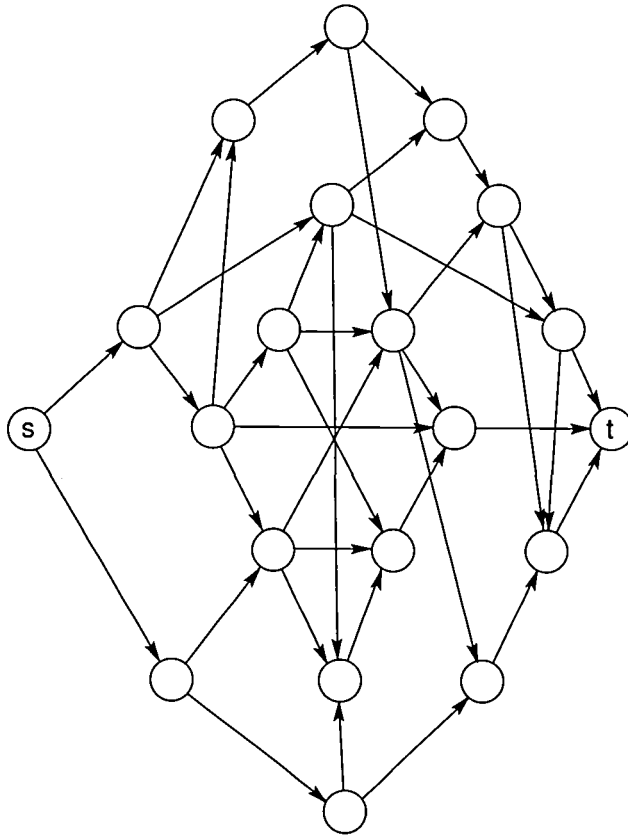


Figure 35. Large Kleindorfer network (ref. 14).

Comparison Between Structural Factoring and Complete Enumeration

The effort required to solve the examples presented in the previous sections with structural factoring is compared in table 6 with the effort required with complete enumeration. As the size of the network in terms of number of nodes and arcs increases, the total number of subnetworks and cpu time increases in an exponential manner as expected for an NP-complete problem. However, the structural factoring approach is dramatically superior to the straightforward complete enumeration approach

and allows exact solution to large network problems that were previously analyzed only by approximation techniques.

Conclusions

The structural factoring approach allows the exact distribution of the shortest path length through a stochastic network to be determined. Application of structural factoring to fan, loop, general loop, and double loop constructs significantly increases the tractability and applicability of the approach, since complete enumeration of all states is not necessary to simplify these constructs. Although there are network architectures that do not initially contain these special constructs, factoring can often be applied locally to produce subnetworks that do contain such constructs. Hence, this approach extends the computational range, especially for cyclic networks, to larger and more complex networks than have previously been solved with other exact methods. Structural factoring can also be applied to find the distribution of the critical path length in acyclic networks and the two-terminal reliability for a network.

Although the technique of structural factoring is theoretically unconstrained, the size of the network that can be solved with the algorithm is restricted by computational resources. In the worst case, the computational effort can grow exponentially with the size of the problem, which is indicative of the NP-complete characterization of the stochastic network problem. Computational experience with the structural factoring algorithm has nonetheless been encouraging. Moreover, the ability to solve certain nontrivial problems exactly can serve as a baseline for assessing the accuracy of various approximation schemes proposed to solve larger problems.

NASA Langley Research Center
Hampton, VA 23665-5225
January 3, 1991

Table 6. Comparison Between Structural Factoring and Complete Enumeration

Network (n, m, k) ^a	Structural factoring		Complete enumeration	
	No. of subnetworks	cpu time, sec	No. of states	cpu time, sec
Fulkerson (4,5,3)	4	3.5	243	1.03
Kleindorfer crossing (6,8,5)	31	3.6	390 625	57.21
Shogan (7,15,3)	1528	8.19	14 348 907	6048.89
Large Kleindorfer (20,38,2)	8191	49.21	2.748779×10^{11}	2.8×10^8

^a n is the number of nodes in the network, m is the number of arcs, and k is the number of "lengths" for each arc.

References

1. Ball, Michael O.: Computational Complexity of Network Reliability Analysis: An Overview. *IEEE Trans. Reliab.*, vol. R-35, no. 3, Aug. 1986, pp. 230–239.
2. Hillier, Frederick S.; and Lieberman, Gerald J.: *Introduction to Operations Research*, 3rd ed. Holden-Day, Inc., c.1980.
3. Dodin, Bajis: Approximating the Distribution Functions in Stochastic Networks. *Comput. & Oper. Res.*, vol. 12, no. 3, 1985, pp. 251–264.
4. Elmaghraby, Salah E.: *Activity Networks: Project Planning and Control by Network Models*. John Wiley & Sons, Inc., c.1977.
5. Frank, H.: Shortest Paths in Probabilistic Graphs, *Oper. Res.*, vol. 17, no. 4, July–Aug. 1969, pp. 583–599.
6. Fulkerson, D. R.: Expected Critical Path Lengths in PERT Networks. *Oper. Res.*, vol. 10, no. 6, Nov.–Dec. 1962, pp. 808–817.
7. Hänsler, E.; McAuliffe, G. K.; and Wilkov, R. S.: Exact Calculation of Computer Network Reliability. *Networks*, vol. 4, 1974, pp. 95–112.
8. Larson, Richard C.; and Odoni, Amedeo R.: *Urban Operations Research*. Prentice-Hall, Inc., c.1981.
9. Martin, J. J.: Distribution of the Time Through a Directed, Acyclic Network. *Oper. Res.*, vol. 13, no. 1, Jan.–Feb. 1965, pp. 46–66.
10. Page, Lavon B.; and Perry, Jo Ellen: Reliability of Directed Networks Using the Factoring Theorem. *IEEE Trans. Reliab.*, vol. 38, no. 5, Dec. 1989, pp. 556–562.
11. Sigal, C. Elliott; Pritsker, A. Alan B.; and Solberg, James J.: The Stochastic Shortest Route Problem. *Oper. Res.*, vol. 28, no. 5, Sept.–Oct. 1980, pp. 1122–1129.
12. Mirchandani, Pitu B.: Shortest Distance and Reliability of Probabilistic Networks. *Comput. & Oper. Res.*, vol. 3, no. 4, 1976, pp. 347–355.
13. Colbourn, Charles J.: *The Combinatorics of Network Reliability*. Oxford Univ. Press, 1987.
14. Kleindorfer, George B.: Bounding Distributions for a Stochastic Acyclic Network. *Oper. Res.*, vol. 19, 1971, pp. 1586–1601.
15. Shogan, A. W.: Bounding Distributions for a Stochastic PERT Network. *Networks*, vol. 7, no. 2, 1977, pp. 359–381.
16. Shogan, Andrew W.: A Recursive Algorithm for Bounding Network Reliability. *IEEE Trans. Reliab.*, vol. R-26, no. 5, Dec. 1977, pp. 322–327.

1. Report No. NASA TP-3069		2. Government Accession No.		3. Recipient's Catalog No.	
4. Title and Subtitle Structural Factoring Approach for Analyzing Stochastic Networks				5. Report Date March 1991	
				6. Performing Organization Code	
7. Author(s) Kelly J. Hayhurst and Douglas R. Shier				8. Performing Organization Report No. L-16794	
				10. Work Unit No. 505-66-21-01	
9. Performing Organization Name and Address NASA Langley Research Center Hampton, VA 23665-5225				11. Contract or Grant No.	
				13. Type of Report and Period Covered Technical Paper	
12. Sponsoring Agency Name and Address National Aeronautics and Space Administration Washington, DC 20546-0001				14. Sponsoring Agency Code	
15. Supplementary Notes Kelly J. Hayhurst: Langley Research Center, Hampton, Virginia. Douglas R. Shier: The College of William and Mary, Williamsburg, Virginia.					
16. Abstract In this report, the problem of finding the distribution of the shortest path length through a stochastic network is investigated. A general algorithm is developed for determining the exact distribution of the shortest path length. The algorithm is based on the concept of conditional factoring, in which a directed, stochastic network is decomposed into an equivalent set of smaller, generally less complex subnetworks. Several network constructs are identified and exploited so that the computational effort required to solve a network problem is significantly less than that required by complete enumeration. This algorithm can be applied to two important classes of stochastic path problems: determining the critical path distribution for acyclic networks and the exact two-terminal reliability for probabilistic networks. Computational experience with the algorithm has been encouraging and has allowed the exact solution of networks previously analyzed only by approximation techniques.					
17. Key Words (Suggested by Authors(s)) Networks Two-terminal reliability Conditional factoring Shortest path				18. Distribution Statement Unclassified—Unlimited	
				Subject Category 66	
19. Security Classif. (of this report) Unclassified		20. Security Classif. (of this page) Unclassified		21. No. of Pages 22	
				22. Price A03	

